US008140975B2

(12) **United States Patent**
Lemay et al.

(10) **Patent No.: US 8,140,975 B2**
(45) **Date of Patent: Mar. 20, 2012**

(54) **SLIDE SHOW NAVIGATION**

(75) Inventors: **Stephen Lemay**, San Francisco, CA (US); **Gregory N. Christie**, San Jose, CA (US); **Marcel van Os**, San Francisco, CA (US); **Hans-Werner Neubrand**, Cupertino, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 572 days.

(21) Appl. No.: **11/319,906**

(22) Filed: **Dec. 27, 2005**

(65) **Prior Publication Data**

US 2006/0156240 A1 Jul. 13, 2006

Related U.S. Application Data

(60) Provisional application No. 60/642,153, filed on Jan. 7, 2005.

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/730**; 715/838

(58) **Field of Classification Search** 715/730, 715/732, 210, 211, 251, 851; 709/206
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,752,893	A	6/1988	Guttag et al.
5,168,441	A	12/1992	Onarheim et al.
5,289,574	A	2/1994	Sawyer
5,297,250	A	3/1994	Leroy et al.
5,357,603	A	10/1994	Parker
5,388,201	A	2/1995	Hourvitz et al.
5,481,665	A	1/1996	Okada et al.
5,490,246	A	2/1996	Brotsky et al.
5,522,022	A	5/1996	Rao et al.
5,537,630	A	7/1996	Berry et al.
5,588,098	A	12/1996	Chen et al.
5,602,997	A	2/1997	Carpenter et al.
5,638,501	A	6/1997	Gough et al.
5,651,107	A	7/1997	Frank et al.
5,657,049	A	8/1997	Ludolph et al.
5,731,819	A	3/1998	Gagne et al.
5,742,285	A	4/1998	Ueda
5,764,229	A	6/1998	Bennett
5,764,238	A	6/1998	Lum et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 548586 A2 6/1993

(Continued)

OTHER PUBLICATIONS

Microsoft Corporation, User's Guide Microsoft® Windows™ and MS-DOS® 6, 1993, pp. Cover-xvi, 112-121.

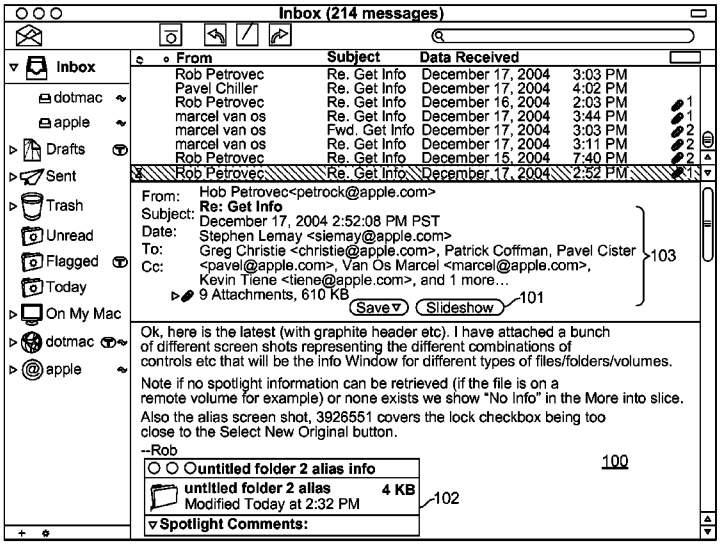
(Continued)

Primary Examiner — William Bashore
Assistant Examiner — Meseker Takele
(74) *Attorney, Agent, or Firm* — Fenwick & West LLP

(57) **ABSTRACT**

A slide show, integrated into an email application, displays images such as those included as an attachment to an email message. The user can view the slide show without exiting the email application. Slide show navigation controls are provided. An index sheet mode can be activated wherein a number of thumbnails are displayed, allowing easy navigation to any image within the slide show. Transition animations can be provided for clarifying operations such as activating and dismissing the index sheet.

46 Claims, 6 Drawing Sheets


Inbox (214 messages)
Inbox
dotmac
apple
Drafts
Sent
Trash
Unread
Flagged
Today
On My Mac
dotmac
apple
From
Subject
Data Received
Rob Petrovec Re. Get Info December 17, 2004 3:03 PM
Pavel Chiller Re. Get Info December 17, 2004 4:02 PM
Rob Petrovec Re. Get Info December 16, 2004 2:03 PM
marcel van os Re. Get Info December 17, 2004 3:44 PM
marcel van os Fwd. Get Info December 17, 2004 3:03 PM
marcel van os Re. Get Info December 17, 2004 3:11 PM
Rob Petrovec Re. Get Info December 15, 2004 7:40 PM
Rob Petrovec Re. Get Info December 17, 2004 2:52 PM
From: Hob Petrovec<petrock@apple.com>
Subject: Re: Get Info
Date: December 17, 2004 2:52:08 PM PST
To: Stephen Lemay <siemay@apple.com>
Cc: Greg Christie <christie@apple.com>, Patrick Coffman, Pavel Cister <pavel@apple.com>, Van Os Marcel <marcel@apple.com>, Kevin Tiene <tiene@apple.com>, and 1 more...
9 Attachments, 610 KB
Save
Slideshow
103
101
Ok, here is the latest (with graphite header etc). I have attached a bunch of different screen shots representing the different combinations of controls etc that will be the Info Window for different types of files/folders/volumes.
Note if no spotlight information can be retrieved (if the file is on a remote volume for example) or none exists we show "No Info" in the More into slice.
Also the alias screen shot, 3926551 covers the lock checkbox being too close to the Select New Original button.
--Rob
100
untitled folder 2 alias info
untitled folder 2 alias 4 KB
Modified Today at 2:32 PM
102
Spotlight Comments:

U.S. PATENT DOCUMENTS

5,793,376	A	8/1998	Tanaka et al.	
5,796,402	A	8/1998	Ellison-Taylor	
5,801,703	A	9/1998	Bowden et al.	
5,835,692	A	11/1998	Cragun et al.	
5,835,693	A	11/1998	Lynch et al.	
5,838,316	A	11/1998	Arruza	
5,877,741	A	3/1999	Chee et al.	
5,877,762	A	3/1999	Young et al.	
5,883,639	A	3/1999	Walton et al.	
5,920,659	A	7/1999	Iverson et al.	
5,933,148	A	8/1999	Oka et al.	
5,949,409	A	9/1999	Tanaka et al.	
5,978,579	A	11/1999	Buxton et al.	
5,999,948	A	12/1999	Nelson et al.	
6,005,568	A	12/1999	Simonoff et al.	
6,006,231	A	12/1999	Popa	
6,011,562	A	1/2000	Gagne et al.	
6,031,937	A	2/2000	Graffagnino	
6,045,446	A	4/2000	Ohshima	
6,075,543	A	6/2000	Akeley	
6,144,381	A	11/2000	Lection et al.	
6,154,601	A	11/2000	Yaegashi et al.	
6,166,748	A	12/2000	Van Hook et al.	
6,167,533	A	12/2000	Potterveld et al.	
6,191,797	B1	2/2001	Politis	
6,195,664	B1	2/2001	Tolfa	
6,211,890	B1	4/2001	Ohba	
6,246,418	B1	6/2001	Oka	
6,266,053	B1	7/2001	French et al.	
6,266,430	B1	7/2001	Rhoads	
6,272,484	B1	8/2001	Martin et al.	
6,272,558	B1	8/2001	Hui et al.	
6,278,448	B1	8/2001	Brown et al.	
6,304,684	B1	10/2001	Niczyporuk et al.	
6,307,574	B1	10/2001	Ashe et al.	
6,310,621	B1	10/2001	Gagne et al.	
6,321,314	B1	11/2001	Van Dyke	
6,353,437	B1	3/2002	Gagne	
6,369,823	B2	4/2002	Ohba	
6,369,830	B1	4/2002	Brunner et al.	
6,396,520	B1	5/2002	Ording	
6,411,301	B1	6/2002	Parikh et al.	
6,412,021	B1	6/2002	Nguyen	
6,421,058	B2	7/2002	Parikh et al.	
6,424,348	B2	7/2002	Parikh	
6,452,600	B1	9/2002	Parikh et al.	
6,456,290	B2	9/2002	Parikh et al.	
6,457,034	B1	9/2002	Morein	
6,466,218	B2	10/2002	Parikh et al.	
6,469,714	B2	10/2002	Buxton et al.	
6,483,524	B1	11/2002	Petchenkine et al.	
6,489,963	B2	12/2002	Parikh et al.	
6,512,522	B1	1/2003	Miller et al.	
6,525,736	B1	2/2003	Erikawa et al.	
6,526,174	B1	2/2003	Graffagnino	
6,542,160	B1	4/2003	Abgrall	
6,544,295	B1	4/2003	Bodnar	
6,571,328	B2	5/2003	Liao et al.	
6,573,896	B1	6/2003	Ribadeau et al.	
6,577,317	B1	6/2003	Duluk, Jr. et al.	
6,580,430	B1	6/2003	Hollis	
6,590,592	B1	7/2003	Nason et al.	
6,609,977	B1	8/2003	Shimizu et al.	
6,614,444	B1	9/2003	Duluk, Jr. et al.	
6,618,048	B1	9/2003	Leather	
6,636,214	B1	10/2003	Leather et al.	
6,639,595	B1	10/2003	Drebin et al.	
6,664,958	B1	12/2003	Leather et al.	
6,664,962	B1	12/2003	Komsthoeft et al.	
6,664,986	B1	12/2003	Kopelman et al.	
6,674,438	B1	1/2004	Yamamoto et al.	
6,697,074	B2	2/2004	Parikh et al.	
6,707,462	B1	3/2004	Peercy et al.	
6,714,201	B1	3/2004	Grinstein et al.	
6,715,053	B1	3/2004	Grigor	
6,717,599	B1	4/2004	Olano	
6,734,864	B2	5/2004	Abgrall	
6,765,592	B1	7/2004	Pletcher et al.	
6,788,318	B2	9/2004	Chen	
6,806,892	B1	10/2004	Plow et al.	
6,906,720	B2	6/2005	Emberling et al.	
6,910,000	B1	6/2005	Yedidia et al.	
6,911,984	B2	6/2005	Sabella et al.	
6,931,633	B1	8/2005	Vazquez et al.	
6,944,829	B2	9/2005	Dando	
6,993,721	B2	1/2006	Rosin et al.	
7,016,011	B2	3/2006	De Haan	
7,027,055	B2	4/2006	Anderson et al.	
7,028,264	B2	4/2006	Santoro et al.	
7,050,955	B1	5/2006	Carmel et al.	
7,076,730	B1 *	7/2006	Baker	715/210
7,085,994	B2	8/2006	Gvily	
7,102,643	B2 *	9/2006	Moore et al.	345/473
7,127,713	B2	10/2006	Davis et al.	
7,174,512	B2	2/2007	Martin et al.	
7,191,399	B2 *	3/2007	Ohtani et al.	715/251
7,243,298	B2 *	7/2007	Yozell-Epstein et al.	715/273
7,281,202	B2	10/2007	Croney et al.	
7,315,848	B2	1/2008	Pearse et al.	
2001/0030647	A1	10/2001	Sowizral et al.	
2001/0035885	A1	11/2001	Iron et al.	
2002/0065949	A1	5/2002	Heaton	
2002/0067418	A1	6/2002	I	
2002/0089526	A1	7/2002	Buxton et al.	
2002/0093516	A1	7/2002	Brunner et al.	
2002/0099678	A1	7/2002	Albright et al.	
2002/0112180	A1	8/2002	Land et al.	
2002/0118217	A1	8/2002	Fujiki	
2002/0120673	A1	8/2002	Tolson et al.	
2002/0129092	A1	9/2002	Tolson et al.	
2002/0171682	A1	11/2002	Frank et al.	
2002/0174003	A1	11/2002	Redmann et al.	
2002/0174181	A1	11/2002	Wei	
2002/0186257	A1	12/2002	Cadiz et al.	
2003/0009267	A1	1/2003	Dunsky et al.	
2003/0020671	A1	1/2003	Santoro et al.	
2003/0046316	A1	3/2003	Gergic et al.	
2003/0080995	A1	5/2003	Tenenbaum et al.	
2003/0122863	A1 *	7/2003	Dieberger et al.	345/730
2003/0123739	A1	7/2003	Graffagnino	
2003/0146934	A1	8/2003	Bailey et al.	
2003/0154239	A1	8/2003	Davis et al.	
2003/0158975	A1	8/2003	Frank et al.	
2003/0160814	A1 *	8/2003	Brown	345/732
2003/0164862	A1	9/2003	Cadiz et al.	
2003/0174136	A1	9/2003	Emberling et al.	
2003/0174154	A1	9/2003	Yukie et al.	
2003/0189597	A1	10/2003	Anderson et al.	
2003/0191799	A1	10/2003	Araujo et al.	
2004/0012626	A1	1/2004	Brookins	
2004/0032409	A1	2/2004	Girard	
2004/0036711	A1	2/2004	Anderson	
2004/0039934	A1	2/2004	Land et al.	
2004/0125128	A1 *	7/2004	Chang et al.	345/730
2004/0179019	A1	9/2004	Sabella et al.	
2004/0212640	A1	10/2004	Mann et al.	
2004/0215740	A1	10/2004	Frank et al.	
2004/0223003	A1	11/2004	Heirich et al.	
2004/0255253	A1	12/2004	Marcjan	
2004/0261012	A1	12/2004	Balsiger	
2004/0261037	A1	12/2004	Ording et al.	
2004/0261038	A1	12/2004	Ording et al.	
2005/0010634	A1	1/2005	Henderson et al.	
2005/0021935	A1	1/2005	Schillings et al.	
2005/0022139	A1	1/2005	Gettman et al.	
2005/0039144	A1	2/2005	Wada et al.	
2005/0057497	A1	3/2005	Kawahara	
2005/0060655	A1	3/2005	Gray et al.	
2005/0060661	A1	3/2005	Kawahara et al.	
2005/0088447	A1	4/2005	Hanggie et al.	
2005/0088452	A1	4/2005	Hanggie et al.	
2005/0144563	A1	6/2005	Hough et al.	
2005/0168471	A1	8/2005	Paquette	
2005/0168476	A1	8/2005	Levene et al.	
2005/0193368	A1	9/2005	Becker et al.	
2005/0240857	A1	10/2005	Benedict et al.	
2005/0256940	A1	11/2005	Henderson et al.	

2005/0278651	A1	12/2005	Coe et al.	
2005/0283734	A1	12/2005	Santoro et al.	
2006/0075033	A1*	4/2006	Bienstock et al.	709/206
2006/0075141	A1	4/2006	Boxenhorn	
2006/0123356	A1	6/2006	Sobeski et al.	
2006/0206835	A1	9/2006	Chaudhri et al.	
2006/0274086	A1	12/2006	Forstall et al.	
2006/0277469	A1	12/2006	Chaudhri et al.	
2006/0277481	A1	12/2006	Forstall et al.	
2007/0038934	A1	2/2007	Fellman	
2007/0044039	A1	2/2007	Amadio et al.	
2007/0061724	A1	3/2007	Slothouber et al.	
2007/0101146	A1	5/2007	Louch et al.	
2007/0101279	A1	5/2007	Chaudhri et al.	
2007/0101288	A1	5/2007	Forstall et al.	
2007/0101291	A1	5/2007	Forstall et al.	
2007/0101297	A1	5/2007	Forstall et al.	
2007/0101433	A1	5/2007	Louch et al.	
2007/0118813	A1	5/2007	Forstall et al.	
2007/0130541	A1	6/2007	Louch et al.	
2007/0162850	A1	7/2007	Adler et al.	
2007/0203984	A2	8/2007	AlHusseini et al.	
2007/0209013	A1	9/2007	Ramsey et al.	
2007/0233736	A1	10/2007	Xiong et al.	
2007/0266093	A1	11/2007	Forstall et al.	
2008/0016491	A1*	1/2008	Doepke	717/104
2009/0177662	A1*	7/2009	Ryder	707/10
2010/0082680	A1*	4/2010	Ryder	707/783
2010/0083351	A1*	4/2010	Ryder	726/4

FOREIGN PATENT DOCUMENTS

EP	0694879	A2	1/1996
EP	1383080	A1	1/2004
EP	0972273	B1	3/2004
WO	WO 98/45815		10/1998
WO	WO 02/09039	A2	1/2002
WO	WO 2004/027707	A2	4/2004

OTHER PUBLICATIONS

Microsoft Corporation, Microsoft® Windows™ User's Guide for the Windows Graphical Environment, Version 3.0 for the MS-DOS® or PC-DOS Operating System, 1990, Document No. SY06851-0290, pp. Cover-vii, 15-76, 315-354.

Conner, D., et al. "Three-Dimensional Widgets," ACM, 1992, pp. 183-231.

"Stardock News: DesktopX User Manual On-line," 1999, 2003, [online] [Retrieved on May 11, 2007] Retrieved from the Internet <URL:http://www.stardock.com/newsitem.asp?id=538>.

"Comparison of widget engines" Wikipedia, 2007 [online] [Retrieved on Sep. 28, 2007] Retrieved from the Internet <URL: http://en.wikipedia.org/wiki/Comparison_of_widget_engines>.

"Dashboard Blog," Dec. 2003 [online] [Retrieved on May 11, 2007] Retrieved from the Internet <URL:http://www.nat.org/dashboard/blog.php3>.

Stardock.com, et al., "DesktopX Whitepaper and Users Guide," Stardock.com, et al., 1999 [online] [Retrieved on May 14, 2007] Retrieved from the Internet <URL:http://www.stardock.net/media/whitepaper_desktopx.html>.

"Windows Sidebar" Wikipedia, 2007 [online] [Retrieved on May 29, 2007] Retrieved from the Internet <URL: http://en.wikipedia.org/wiki/Windows-Sidebar>.

Gruber, J., et al., "Dashboard vs. Konfabulator," Daring Fireball, Jun. 2004, [online] [Retrieved on May 11, 2007] Retrieved from the Internet <URL:http://daringfireball.net/2004/06/dashboard_vs._konfabulator>.

"Snippets Software Platform," Snippet Software Inc., Jun. 2001, [online] [Retrieved on Jun. 11, 2001] Retrieved from the Internet <URL:http://www.snippets.com/products/>.

Snippet Software Inc. et al., "Product Spotlight Non-browser based portal solution from Snippets Software, Inc.," Corporate Portal Letter, Oct. 2000, 3 Pages, vol. 1, No. 10.

International Search Report and Written Opinion, PCT/US2005/022152, Jul. 10, 2006, 8 pages.

Archive of "Objects, Images and Applets," W3C Recommendation, Feb. 10, 2003, [online] [Archived by http://archive.org; Retrieved on Apr. 13, 2006] Retrieved from the Internet<URL:http://web.archive.org/web/20030210154019/http://www.w3.org/TR/REC-html140/struct/objects.html>.

Ullenboom, C., "Java is auch eine Insel," 2003, [online] [Retrieved on Apr. 13, 2006] Retrieved from the Internet<URL:http://www.galileocomputing.de/openbook/javainsel2//java_140000.htm#Xxx998138>.

Akeley, Kurt et al., "Real-Time Graphics Architecture," The OpenGL® Graphics System, CS448 Lecture 15 [online] Fall, 2001Retrieved from the Internet: <URL: http://www.graphics.stanford.edu/courses/cs448a-01-fall>, pp. 1-20.

Cadiz, JJ et al., "Sideshow: Providing Peripheral Awareness of Important Information," Technical Report MSR-TR-2001-83, Sep. 14, 2001, Microsoft Corporation, Redmond, WA, 9 pages.

Elliott, Conal, "Programming Graphics Processors Functionally," 11 pages.

Fried, Ina, "Developer Calls Apple's Tiger a Copycat," CNET News.com, Jun. 28, 2004 [online] [Retrieved on Jul. 1, 2004] Retrieved from the Internet<URL:http://zdnet.com.com/2102-1104_2-250692.html?tag=printthis>.

Fried, Ina, "For Apple's Tiger, the Keyword is Search," CNET News.com, Jun. 28, 2004 [online] [Retrieved on Jul. 1, 2004] Retrieved from the Internet<URL:http://zdnet.com.com/2102-1103_2-250346.html?tag=printthis>.

Haeberli, P. et al., "The Accumulation Buffer: Hardware Support for High-Quality Rendering," Computer Graphics, Aug. 1990, pp. 309-318, vol. 24, No. 4.

International Search Report, PCT/US2005/008804, Jul. 27, 2005, 3 pages.

International Search Report, PCT/US2005/008805, Aug. 8, 2005, 3 pages.

Konfabulator, "Cupertino, Start Your Photocopiers!," [online] [Retrieved on Jul. 1, 2004] Retrieved from the Internet<URL: http://www.konfabulator.com>.

Konfabulator, Konfabulator & Widget Basics, [online] [Retrieved on Jul. 1, 2004] Retrieved from the Internet<URL: http://www.konfabulator.com/info/basics.html>.

Konfabulator, "Screenshots," [online] [Retrieved on Jul. 1, 2004] Retrieved from the Internet<URL:http://www.konfabulator.com/info/screenshots.html>.

Konfabulator, "What is Konfabulator?," [online] [Retrieved on Jul. 1, 2004] Retrieved from the Internet<URL:http://www.konfabulator.com/info/>.

Notification of Transmittal of the International Search Report and the Written Opinion of the International Searching Authority, PCT/US2005/022579, 14 pages.

NVIDIA, "Cg—Teaching Cg," Power Point Presentation, Author and date unknown, pp. 1-16.

Puder, A., "Extending Desktop Applications to the Web," ACM International Conference Proceedings Series, Proceedings of the 2004 International Symposium on Information and Communication Technologies, 2004, vol. 90, 6 pages.

Rochkind, M. et al., "Common Elements in Today's Graphical User Interfaces: The Good, the Bad, and the Ugly," INTERCHI '93, ACM, Apr. 24-29, 1993, pp. 470-473.

Segal, Mark et al., "The OpenGL® Graphics System: A Specification (Version 1.5)," Silicon Graphics, Inc., Oct. 30, 2003, 334 pages.

Shantzis, Michael A., "A Model for Efficient and Flexible Image Computing," Computer Graphics Proceedings, Annual Conference Series, Jul. 24-29, 1994, pp. 147-154, Orlando, FL.

Shiozawa, Hidekazu et al., "Perspective Layered Visualization of Collaborative Workspaces," Proceedings of the International ACM SIGGROUP conference on Supporting Group Work Publisher, Nov. 1999.

Staples, Loretta, "Representation in Virtual Space: Visual Convention in the Graphical User Interface," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Apr. 1993.

Tang, J.C. et al., "ConNexus to Awarenex: Extending Awareness to Mobile Users," SIGCHI '01, ACM, Mar. 31-Apr. 4, 2001, 8 pages.

Van Gelder, Allen et al., "Direct Volume Rendering with Shading Via Three-Dimensional Textures," University of California, Santa Cruz, CA, 9 pages.

Wardell, Brad, Konfabulator for Windows, Jan. 10, 2004; [online]Retireved from the Internet Mar. 6, 2006] Retrieved from the Internet<URL: http://www.xpthemes.com/forums.asp?MID=19&CMID=19&AID=4472>.

"Writing a Desk Accessory," Developer Connection, Apple Computer, Inc. Jul. 3, 1996, [online] [Retrieved on Jan. 3, 2006] Retrieved from the Internet<URL:http://developer.apple.com/documentation/mac/Devices/Devices-16>.

Snippets Software et al., "Products: Overview," Feb. 2002, [online] [Retrieved on Feb. 5, 2008] Retrieved from the Internet URL:http://web.archive.org/web/20020206061508/http://www.snippets.com/products/>.

Stardock et al., "DesktopX General Guide," Aug. 2000, [online] [Retrieved on Jan. 31, 2008] Retrieved from the internet <URL:http://www.stardock.com/products/desktopx/docs/>.

Stardock et al., "DesktopX WhitePaper," Aug. 2000, [online] [Retrieved on Jan. 31, 2008] Retrieved from the internet <URL:http://www.stardock.net/media/whitepaper_desktopx.html>.

Stardock et al., "DesktopX Tutorial," Aug. 2000, [online] [Retrieved on Jan. 31, 2008] Retrieved from the internet <URL:http://www.stardock.com/products/desktopx/tutorial.html>.

Stardock et al., "What Can It Do? Making Objects," DesktopX Tutorial, 2001, [online] [Retrieved on Apr. 11, 2008] Retrieved from the Internet <URL:http://web.archive.org/web/20011019222825/http://www.stardock.com/products/desktopx/...>.

Altman, R. B., "Applying Preset Animations" in Visual QuickStart Guide Power Point 2000/98, ProQuest Safari Books, Peachpit Press, May 7, 1999, 7 pages [online] Retrieved from the Internet: <URL: http://proquest.safaribooksonline.com/0201354411>.

Lammers, J., et al., "Maya 4.5 Fundamentals: Particles," New Riders Publishing, Jan. 14, 2003, 12 pages, [online] [retrieved on Feb. 17, 2007] Retrieved from ProQuest Safari Books Online on the Internet: <URL: http://proquest.safaribooksonline.com/0735713278>.

* cited by examiner

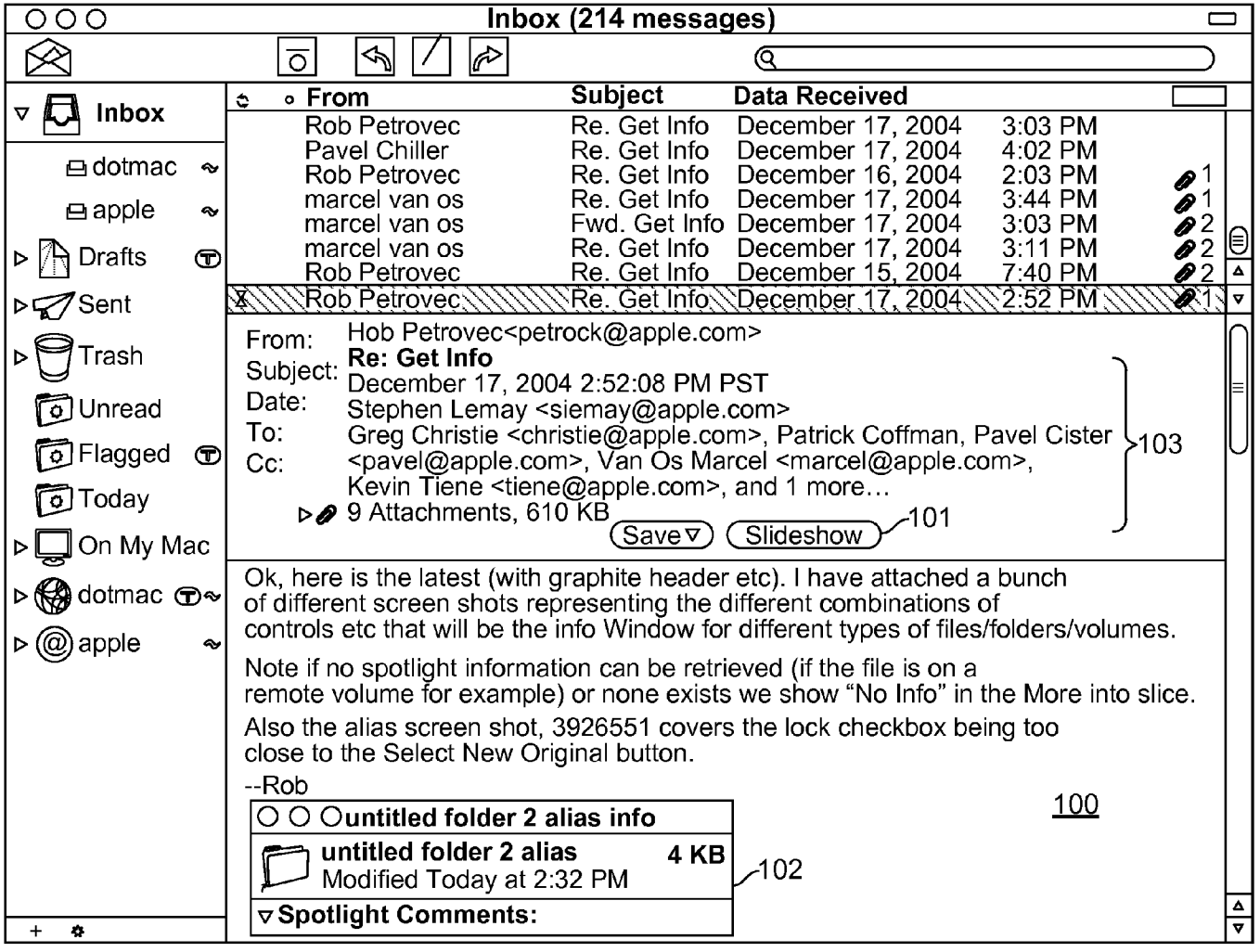

Inbox (214 messages)
Inbox
dotmac
apple
Drafts
Sent
Trash
Unread
Flagged
Today
On My Mac
dotmac
apple
From
Subject
Data Received
Rob Petrovec Re. Get Info December 17, 2004 3:03 PM
Pavel Chiller Re. Get Info December 17, 2004 4:02 PM
Rob Petrovec Re. Get Info December 16, 2004 2:03 PM 1
marcel van os Re. Get Info December 17, 2004 3:44 PM 1
marcel van os Fwd. Get Info December 17, 2004 3:03 PM 2
marcel van os Re. Get Info December 17, 2004 3:11 PM 2
Rob Petrovec Re. Get Info December 15, 2004 7:40 PM 2
Rob Petrovec Re. Get Info December 17, 2004 2:52 PM 1
From: Hob Petrovec<petrock@apple.com>
Subject: Re: Get Info
Date: December 17, 2004 2:52:08 PM PST
To: Stephen Lemay <siemay@apple.com>
Cc: Greg Christie <christie@apple.com>, Patrick Coffman, Pavel Cister <pavel@apple.com>, Van Os Marcel <marcel@apple.com>, Kevin Tiene <tiene@apple.com>, and 1 more…
103
9 Attachments, 610 KB
Save
Slideshow
101
Ok, here is the latest (with graphite header etc). I have attached a bunch of different screen shots representing the different combinations of controls etc that will be the info Window for different types of files/folders/volumes.
Note if no spotlight information can be retrieved (if the file is on a remote volume for example) or none exists we show “No Info” in the More into slice.
Also the alias screen shot, 3926551 covers the lock checkbox being too close to the Select New Original button.
--Rob
100
untitled folder 2 alias info
untitled folder 2 alias 4 KB
Modified Today at 2:32 PM
102
Spotlight Comments:

FIG. 1

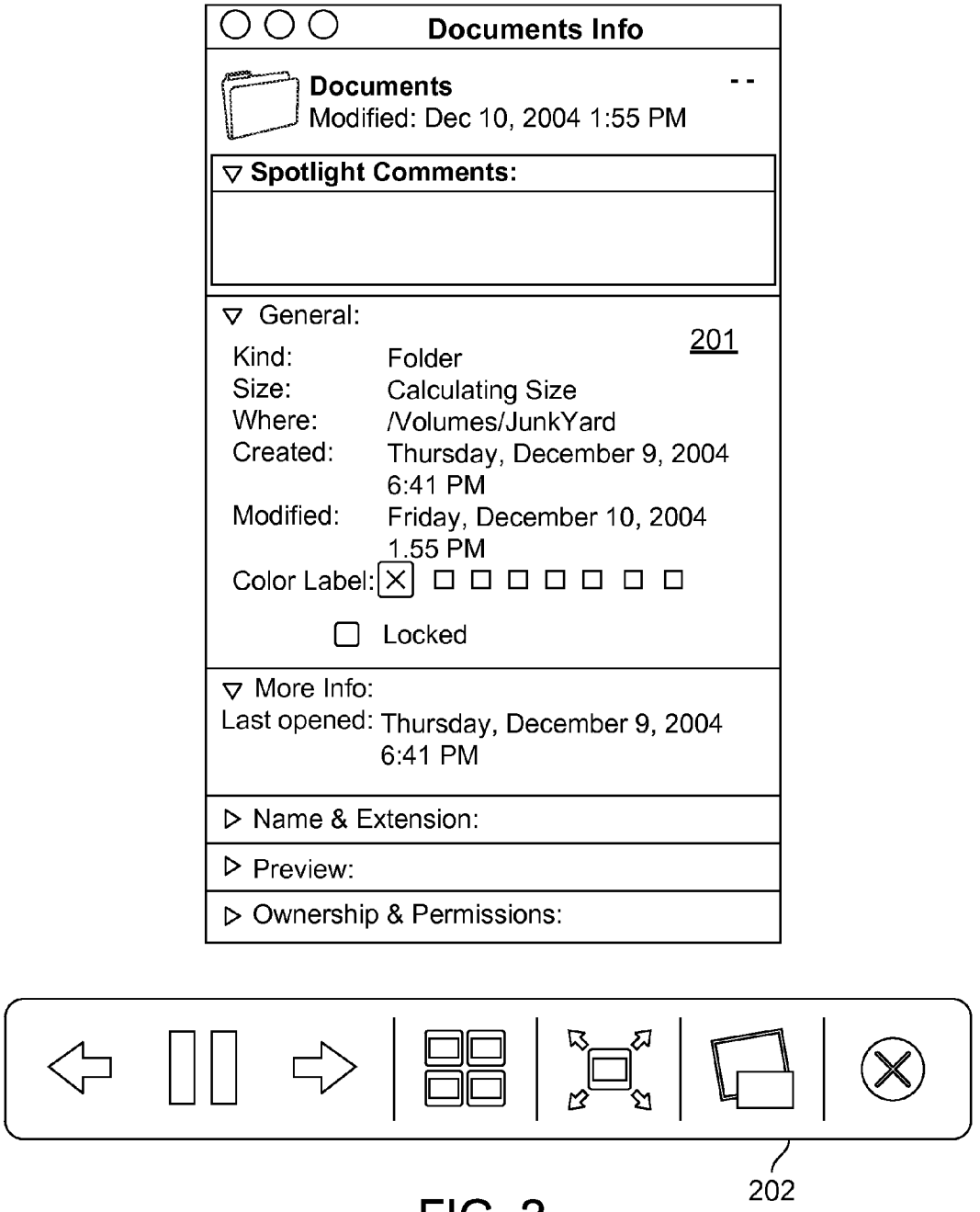

Documents Info
Documents
Modified: Dec 10, 2004 1:55 PM
Spotlight Comments:
General:
201
Kind: Folder
Size: Calculating Size
Where: /Volumes/JunkYard
Created: Thursday, December 9, 2004 6:41 PM
Modified: Friday, December 10, 2004 1:55 PM
Color Label:
Locked
More Info:
Last opened: Thursday, December 9, 2004 6:41 PM
Name & Extension:
Preview:
Ownership & Permissions:
202

FIG. 2

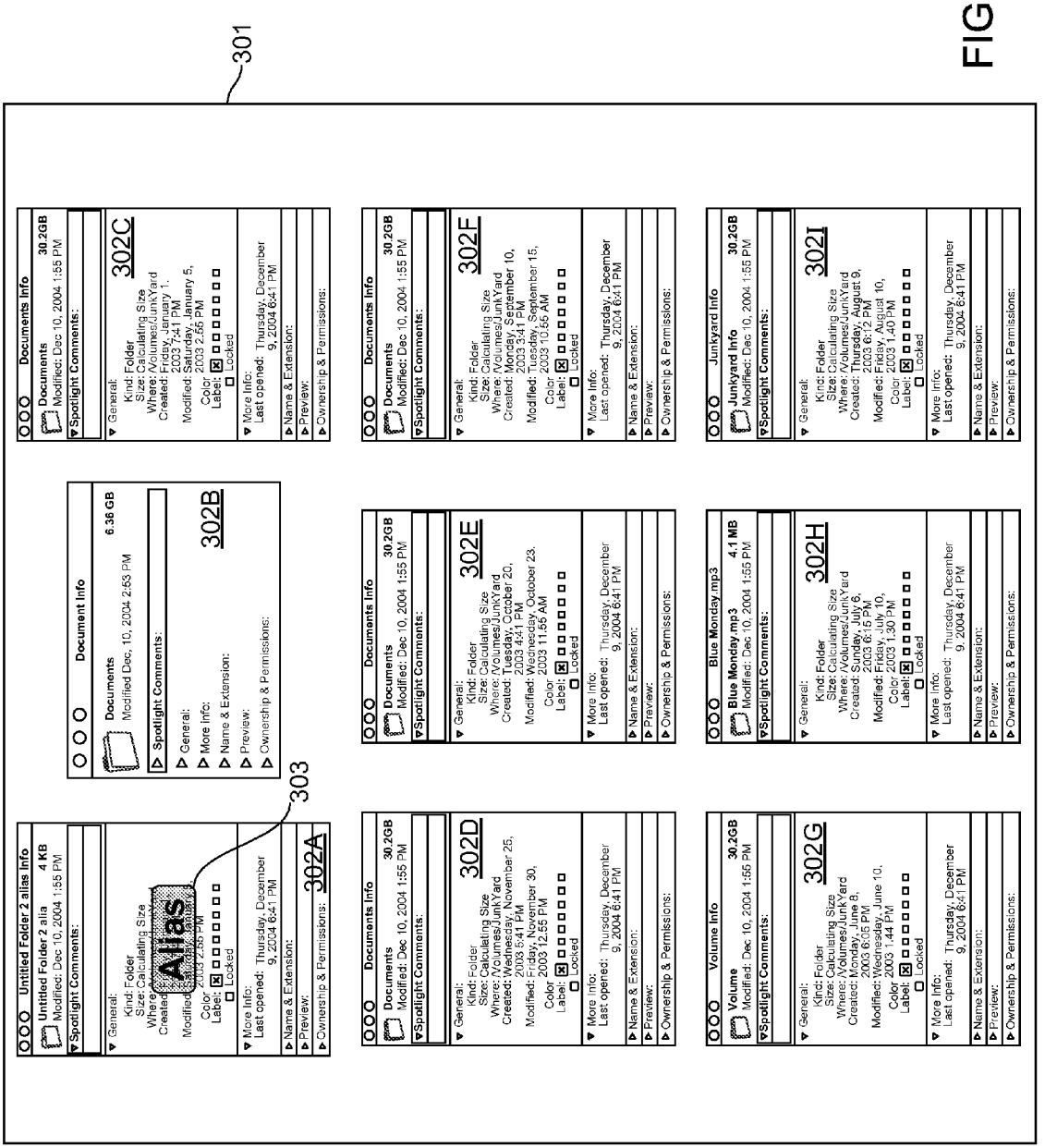


FIG. 3

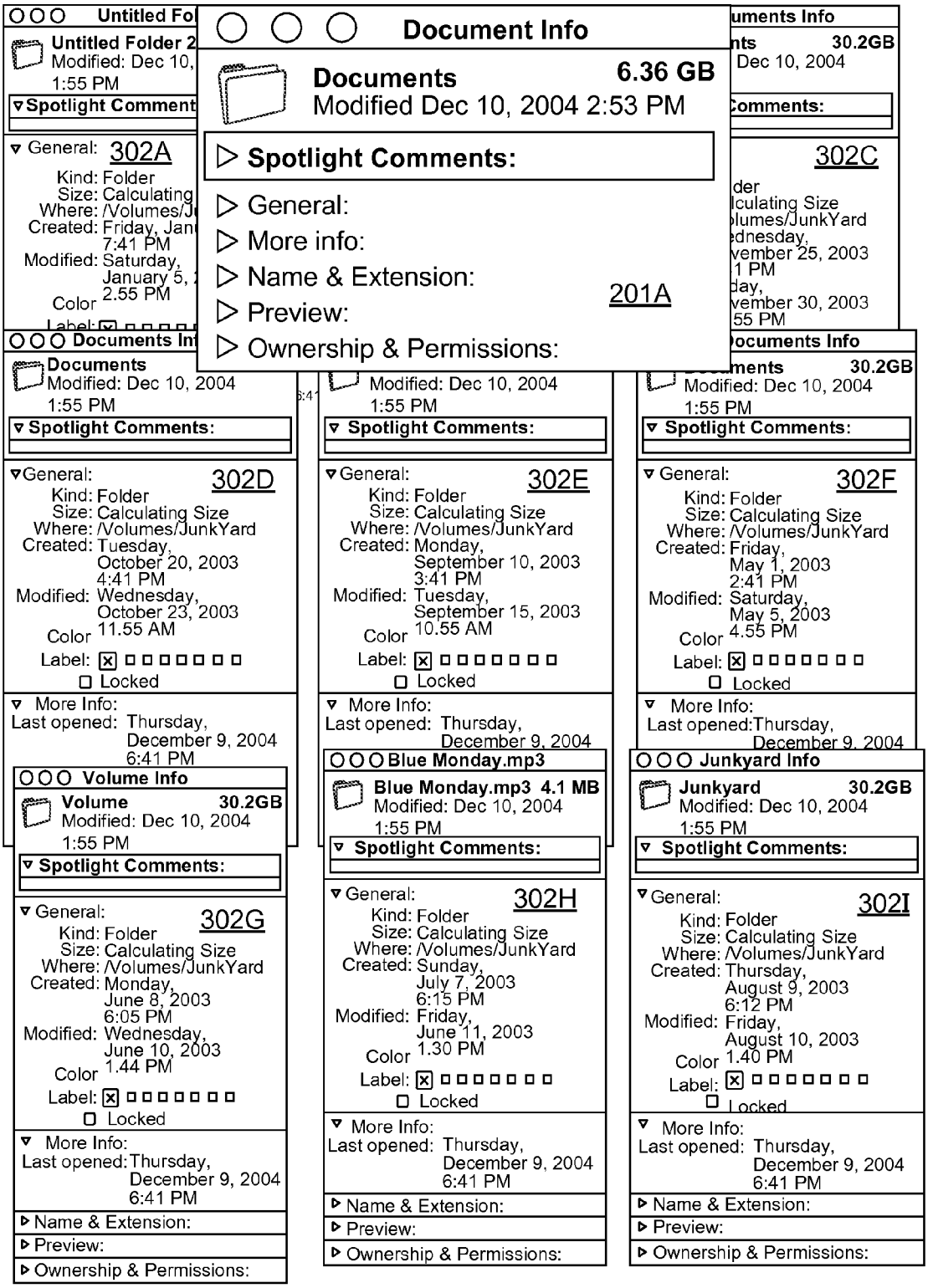

Document Info
Documents
6.36 GB
Modified Dec 10, 2004 2:53 PM
Spotlight Comments:
General:
More info:
Name & Extension:
Preview:
Ownership & Permissions:
201A
302A
302C
302D
302E
302F
302G
302H
302I
Volume Info
Blue Monday.mp3
Junkyard Info
Spotlight Comments:
General:
More Info:
Name & Extension:
Preview:
Ownership & Permissions:

FIG. 4

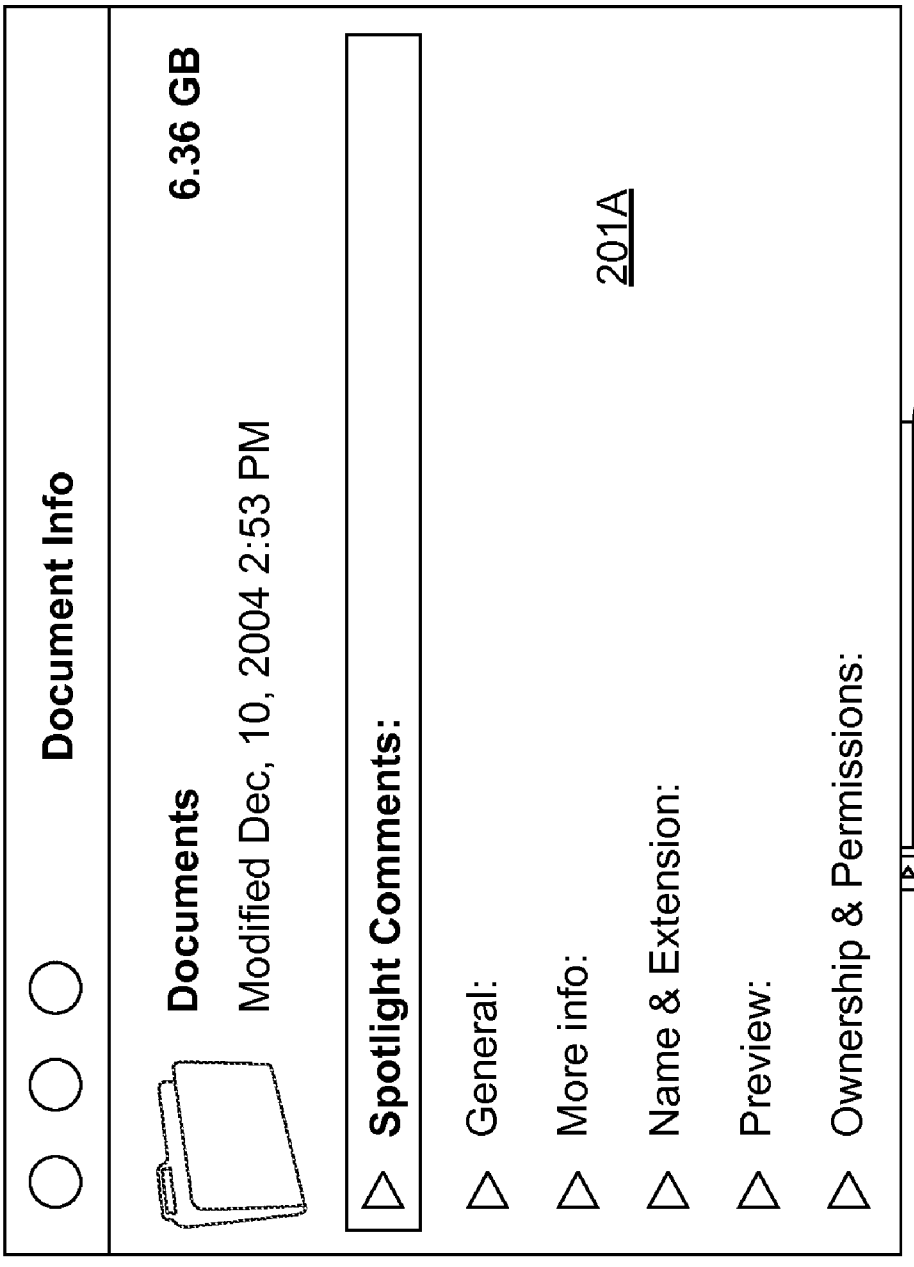

Document Info
Documents
6.36 GB
Modified Dec, 10, 2004 2:53 PM
Spotlight Comments:
General:
More info:
201A
Name & Extension:
Preview:
Ownership & Permissions:
Ownership & Permissions:
302A

FIG. 5

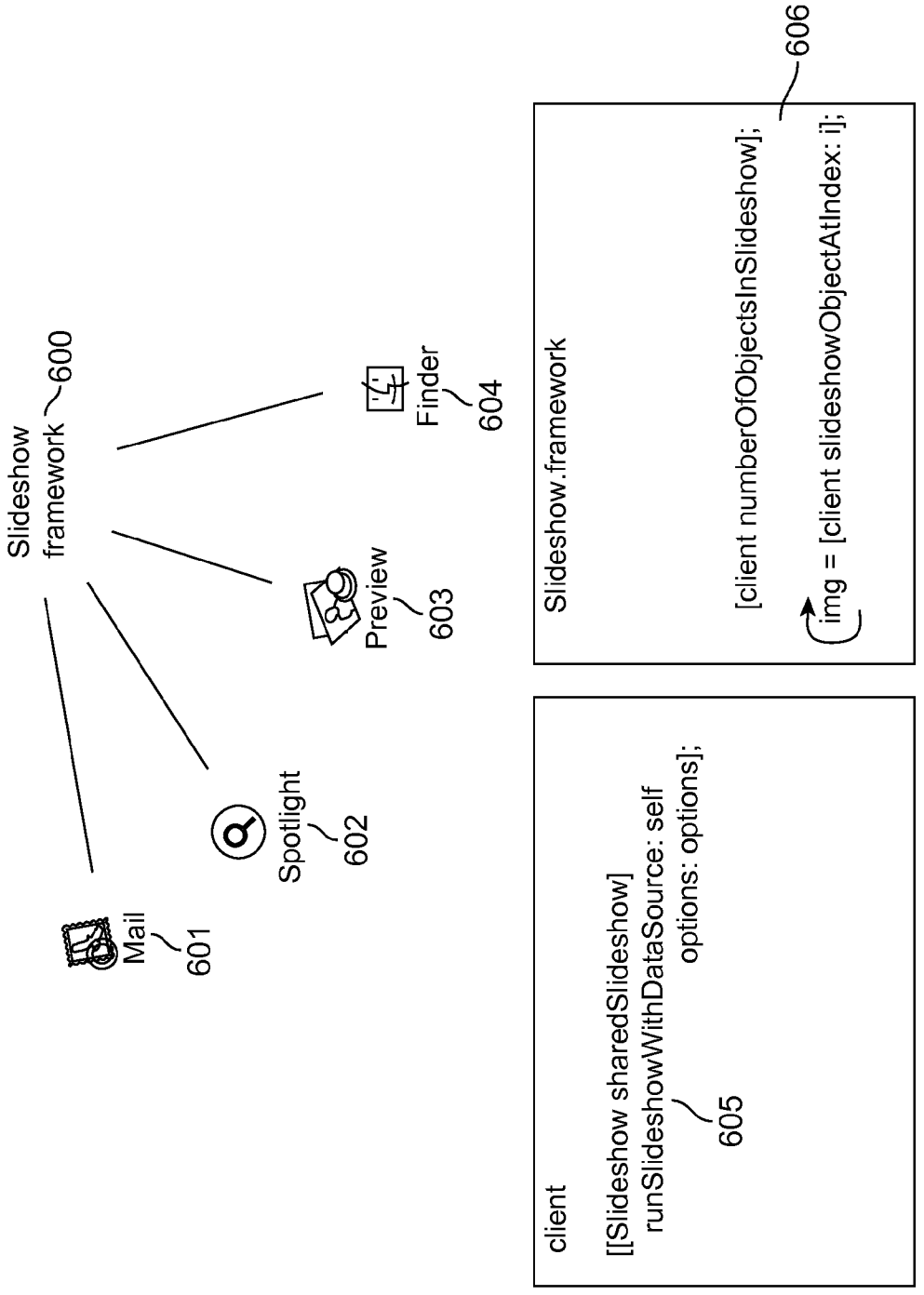

Slideshow framework
600
Mail
601
Spotlight
602
Preview
603
Finder
604
client
[[Slideshow sharedSlideshow]
runSlideshowWithDataSource: self
options: options];
605
Slideshow.framework
[client numberOfObjectsInSlideshow];
img = [client slideshowObjectAtIndex: i];
606

FIG. 6

SLIDE SHOW NAVIGATION

CROSS-REFERENCE TO RELATED APPLICATIONS

The present invention claims priority from the following provisional patent application, the disclosure of which is incorporated herein by reference:

U.S. Provisional Patent Application Ser. No. 60/642,153 for "Slide Show Navigation," filed Jan. 7, 2005.

The present invention is related to the following pending utility patent applications, the disclosures of which are incorporated herein by reference:

U.S. patent application Ser. No. 10/877,968 for "Unified Interest Layer for User Interface," filed Jun. 24, 2004;

U.S. patent application Ser. No. 10/465,855 for "Computer Interface Having a Virtual Single-Layer Mode for Viewing Overlapping Objects," filed Jun. 20, 2003;

U.S. patent application Ser. No. 10/689,687 for "Computer Interface Having a Virtual Single-Layer Mode for Viewing Overlapping Objects," filed Oct. 22, 2003.

FIELD OF THE INVENTION

The present invention relates generally to displaying images within the context of an application such as an email application, and more particularly to improved techniques for navigating within a series of images such as a slide show.

BACKGROUND OF THE INVENTION

Conventionally, there is no convenient way to browse and view a series of image attachments received via email without leaving that email application. Images can appear inline in the body of the email, but browsing through a series of image attachments is restricted to vertically scrolling the body of the email. The presentation size of these image attachments is also restricted to the size of the email application view window, which is typically small. Existing software applications that allow a user to browse and view images, are not generally integrated into email applications. Additionally, a user may have to download and import the images into such software applications in order to view them, but this takes time and is often not desired when the user just wants to quickly see the images in a received email message.

In addition, it is often difficult to navigate within a series of images such as within a slide show. An image may be shown, with arrows for moving forwards and backwards. Such a display does not facilitate instantaneous movement from one image to a second image that is not adjacent to the first. Such a display also fails to provide the user with a sense of the context of the current image within the overall slide show or set of images. There is no easy and convenient way to switch between the full-sized view and a view that reveals context.

Image navigation systems and software often provide a series of thumbnails, allowing the user to click on a thumbnail to see a full-sized version of the corresponding image. However, these schemes still provide no easy way to easily switch between a full-sized view and a view that provides context showing where the image fits within the overall slide show.

SUMMARY OF THE INVENTION

The present invention provides an improved mechanism for navigating within a series of images such as a slide show.

According to one aspect of the present invention, image attachments can be easily viewed without leaving an email application.

According to one aspect of the present invention, images are presented in a manner that provides for easy switching between full-sized and thumbnail view, and that preserves a sense of context so that the user knows where the image fits within the overall slide show. The user can easily switch between modes, and can easily move to any image within the slide show or image series.

According to one aspect of the present invention, the invention provides a full-screen slide show environment, with easy access to instantaneous navigation controls and thumbnails, all accessible directly within an email application.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

FIG. **1** is a screen shot depicting an example of an email client application with a slide show button, according to one embodiment of the present invention.

FIG. **2** is a screen shot depicting an example of an image being displayed in a slide show, according to one embodiment of the present invention.

FIG. **3** is a screen shot depicting an example of an index sheet, according to one embodiment of the present invention.

FIG. **4** is a screen shot depicting an example of a transition that takes place after the user has selected a thumbnail from the index sheet, according to one embodiment of the present invention.

FIG. **5** is a screen shot depicting a continuation of the example transition of FIG. **4**, according to one embodiment of the present invention.

FIG. **6** is a block diagram depicting an example of a software architecture for implementing the present invention according to one embodiment.

One skilled in the art will recognize that these Figures are merely examples of the operation of the invention according to one embodiment, and that other user interface arrangements and modes of operation can be used without departing from the essential characteristics of the invention.

DETAILED DESCRIPTION OF THE EMBODIMENT

The present invention is now described more fully with reference to the accompanying Figures, in which several embodiments of the invention are shown. The present invention may be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather these embodiments are provided so that this disclosure will be complete and will fully convey the invention to those skilled in the art.

For illustrative purposes, the invention is described in connection with presenting images in an email application for an operating system such as MacOS X from Apple Computer, Inc. Various specific details are set forth herein and in the Figures, to aid in understanding the present invention. However, such specific details are intended to be illustrative, and are not intended to restrict in any way the scope of the present invention. In particular, one skilled in the art will recognize that the invention can be used for navigating among images and/or other non-image elements within any type of software application. In addition, one skilled in the art will recognize that the invention can be used in connection with any input mechanism for selecting, manipulating, and moving on-screen images, including manipulation of an on-screen cursor using a pointing device, keyboard control, voice control, mouse wheel, touch screen control, or the like, or any combination of the above. In addition, the particular screen

layouts, appearance, and terminology as depicted and described herein, are intended to be illustrative and exemplary, and in no way limit the scope of the invention.

In one embodiment, the present invention is implemented in a conventional personal computer system, such as a Macintosh, (available from Apple Computer, Inc. of Cupertino, Calif.), running an operating system such as MacOS X (also available from Apple Computer, Inc.). The invention may be embodied in software that runs on the personal computer. The invention may be included as add-on software, or it may form part of the operating system itself, or it may be a feature of an application that is bundled with the computer system or sold separately. The various features of the invention as described herein include output presented on a display screen that is connected to the personal computer. In addition, the invention makes use of input provided to the computer system via input devices such as a keyboard, mouse, touchpad, or the like. Such hardware components, including their operation and interactions with one another and with a central processing unit of the personal computer, are well known in the art of computer systems and therefore are not depicted here. Several Figures are presented as screen shots depicting examples of the user interface as it might appear on a display screen or other output device. In addition, the techniques described herein can be implemented in other types of devices, including cell phones, personal digital assistants, television displays, kiosks, or the like.

User Experience

In one embodiment, the invention operates as follows:

Referring now to FIG. **1**, there is shown a screen shot depicting an example of an email client application displaying a message **100**. The user receives email message **100** that includes one or more image attachments **102**. Slide show button **101** appears in header region **103** of message **100**. In one embodiment, button **101** is only presented if message **100** contains images. In other embodiments, other mechanisms for activating a slide show may be provided, including keyboard commands, menu items, or the like.

The user can click on button **101** to initiate a slide show. The computer screen fades to black and then fades into a full screen view of the first image attachment. Referring now to FIG. **2**, there is shown a screen shot depicting an example of an image **201** being displayed in a slide show, according to one embodiment of the present invention.

After a period (for example 3 seconds), the first image **201** fades into the next image attachment in the sequence, and so on until all the images have been viewed. Then, in one embodiment, the slide show repeats.

In an alternative embodiment, the slide show is presented within a window rather than occupying the entire screen. In another alternative embodiment, the slide show is presented in semi-transparent form overlaying the normal desktop display. In one embodiment, any or all of the characteristics of the slide show are configurable by the user (for example, whether it repeats, the size of the presentation, delay between one slide and the next, and the like).

Semi-transparent toolbar cluster **202** (or dock) appears at the bottom of the screen. It contains a series of buttons that allow the user to interact with the slide show. From left to right they are:

Back: Go to the previous image.

Play/Pause: Animate the slideshow from one image to the next, or pause on the current frame

Next: Go to the next image

Index Sheet: Show all the images in this slideshow.

Full Screen/Actual Size: Toggle between showing the image at its actual size and showing it scaled up to fit the computer screen

Add to iPhoto: Automatically add the current image to iPhoto

Exit Slideshow: Exit the full screen slideshow presentation and return to the application view.

Index Sheet

The Index Sheet button provides quick access to a screen depicting thumbnails. This may be overlaid on the existing desktop, as an opaque screen or a semi-transparent one. In one embodiment, the index sheet can be activated by moving the cursor to a predefined corner of the screen. The index sheet allows a user to easily see the context of the current image within the slide show, and also allows the user to quickly jump to any other image in the slide show.

Referring now to FIG. **3**, there is shown a screen shot depicting an example of an index sheet **301**, according to one embodiment of the present invention.

When the user presses the Index Sheet button in cluster **202** (or otherwise activates the index sheet), the current image **201** animates and scales smaller. At the same time all the other images in the slideshow also animate and scale smaller behind the current image. As the images scale smaller, they align along a position in a grid that reflects the order of the slideshow presentation, as shown in FIG. **3**. The first image appears as a thumbnail **302**A in the upper left, the next image appears as a thumbnail **302**B to the right of the first image, and so on. The spread-out view of all the image thumbnails **302**A through **302**J forms index sheet **301**; it is organized in a rectangular grid akin to a photographic contact sheet that shows a number of images on a single page. In one embodiment, all images are scaled to the same size, or to approximately the same size except for deviations resulting from different aspect ratios. In one embodiment, the size of the thumbnails **302** is selected according to the number of images and the size of the available viewable area. In one embodiment, if there are too many images to show all thumbnails **302** within the viewable area, a scrolling arrangement or other arrangement for viewing subsets of the thumbnails **302** is used. In one embodiment, the size is specified by a user.

In one embodiment, thumbnails **302** are arranged according to the image sequence within the slide show. In one embodiment, the presentation is according to filename (alphabetically), or chronological order by date modified or date created, or size, or some other ordering mechanism. In one embodiment, the user can dynamically select the order.

The user can roll over any thumbnail **302** (in other words, cause the on-screen cursor to hover over the image) to reveal its title **303** and/or to see other information about the image. In one embodiment, such information is shown in a semi-transparent manner superimposed on thumbnail **302** itself, as shown in the example of FIG. **3**. In other embodiments, any other technique for displaying such information can be used. Any type of information can be displayed, including for example title, size, date created, date modified, or the like.

Clicking on a thumbnail **302** advances the slide show to that image and exits Index Sheet mode; the selected image animates and scales to full screen size while all the other images appear to scale to full screen size behind the selected image. Alternatively, the non-selected images may shrink until they are invisible, or they may fade to invisibility, or simply disappear as the selected image expands to fill the screen. Alternatively, they may simply appear to move behind the selected image.

FIGS. **4** and **5** depict an example of the transition between index sheet **301** and a full-screen display of image **201**A. In

FIG. **4**, thumbnails **302**A through **302**J have begun to shift position behind selected image **201**A, which has enlarged to full size. In FIG. **5**, thumbnails **302**A through **302**J have moved so that they are almost completely hidden by selected image **201**A; they are also stacked on top of one another.

One skilled in the art will recognize that any other transition effect (or no transition effect) can be implemented without deviating from the essential characteristics of the present invention.

Add to iPhoto

At any point in the slide show presentation, the user can press the Add to iPhoto button to automatically add the currently viewed image to a photo management application (such as iPhoto, available from Apple Computer of Cupertino, Calif.) without exiting the slide show presentation. In one embodiment, the software displays an animation showing the currently viewed image being sucked into the Add to iPhoto button. This feature allows a user to browse a large series of images, and to easily choose which images he or she would like to import and keep for future use.

The present invention provides a slide show that is accessible from and works within an email application. As such, it does not require the user to leave the email environment in order to view their image attachments in an effective, versatile, full-screen, and pleasant way. In other embodiments, the features described above are provided within other types of applications, such as a file management application, presentation software, word processing software, or the like.

Architecture

In one embodiment, the above-described functionality is implemented as a standalone framework, so that any application or client can generate a user experience such as that described above. The present invention can be implemented in connection with images, PDF documents, text documents, or any other items or elements that can be presented visually.

In one embodiment, the invention is implemented as an application programming interface (API) in objC and/or C, using a known technology such as ImageIO to load images of multiple formats. Image formats include, for example, TIFF, JPEG, PNG, GIF, or the like. Image inputs can be provided in any of a number of forms that either contain or point to images. Examples include file paths, URLs, NSFileWrappers, NSImages, CGImageRefs, or the like.

In one embodiment, the invention includes a dynamically created user interface that adapts to environmental conditions and/or client feature sets. For example, in one embodiment, an Add to iPhoto button may be made available only if 1) the client application includes functionality for exporting images to iPhoto (i.e., if such an operation makes sense given the nature of the application), and 2) the iPhoto application is installed; otherwise, the Add to iPhoto button is not shown. In other environments or clients, the user interface may include other elements that are appropriate to those environments. For example, when the slide show of the present invention is being used in connection with an image viewer or search function, an Add to Mail button can be provided within the button cluster; when activated, this button would launch an email application and create a new email message with the selected image(s) attached. One skilled in the art will recognize that in other contexts, other buttons and operations would be applicable.

In one embodiment, the present invention displays various types of transitions. For example, when switching from one image to the next (in play mode, or in response to a user command), a distinctive transition may be shown; similarly, when switching from full-sized image mode to thumbnail mode, another transition may be shown. In one embodiment, these transitions are user-configurable. In one embodiment, they are implemented using a known video animation technology, assuming that the hardware video card supports such technology. In one embodiment, multiple transitions are available and can be applied in any combination.

In one embodiment, a particular type of transition and visual effect is used when switching from full-sized image mode to thumbnail mode and vice versa. An affine transform (translate/scale) is applied to position each image during the index transform and in index sheet mode. In one embodiment, the thumbnail display is presented in a semi-transparent window so that underlying elements (such as the desktop or the full-sized image that was being displayed) are visible through the thumbnail display. One skilled in the art will recognize that such visual enhancements are optional, and that the present invention can be implemented using many different types of visual techniques.

In one embodiment, the Add to iPhoto function is performed using AppleScript. The system of the invention keeps track of the images that a user wants to add to iPhoto. Upon quitting or closing the slide show, iPhoto is automatically launched, and the selected images are added to iPhoto under the direction of an AppleScript script.

Referring now to FIG. **6**, there is shown a block diagram depicting an example of a software architecture for implementing the present invention according to one embodiment. Slideshow framework **600** is software code that implements the invention. Any client application can call upon and interface with framework **600**; examples of such applications include an email application **601**, search application (such as Apple's Spotlight) **602**, viewer application (such as Apple's Preview) **603**, or file management component of an operating system (such as Apple's Finder) **604**.

Also shown is an example of code **605** that can be used by the calling application to invoke the slide show feature. Also shown is an example of code **606** in framework **600** for interfacing with the calling application.

Examples of header files for use in connection with an implementation of the present invention is as follows:

Slideshow.h

```
#import <AppKit/AppKit.h>
#import <ApplicationServices/ApplicationServices.h>
@class SlideshowPrivateVars;
// Slideshow data source
//-------------------------------------------------------
@interface NSObject(SlideshowDataSource)
// required
- (int)numberOfObjectsInSlideshow;
- (id)slideshowObjectAtIndex: (int)index;
// optional
- (NSString*)slideshowObjectNameAtIndex: (int)index;
- (BOOL)canExportObjectAtIndexToiPhoto: (int)index;
- (void)exportObjectsToiPhoto: (NSIndexSet*)indexSet;
@end
```

-continued

```
        // Slideshow
        //----------------------------------------------------
----------
        @interface Slideshow : NSResponder
        {
                @private
                SlideshowPrivateVars * mPrivateData;
        }
        + (Slideshow*)sharedSlideshow;
        + (void)addImageToiPhoto: (id)image;
        // To start the Slideshow for images:
        // – either use setDataSource/startSlideshow
        – (void)setDataSource: (id)dataSource;
        – (void)startSlideshow: (id)sender;
        //     or use runSlideshowWithDataSource (preferred way)
        – (void)runSlideshowWithDataSource:     (id)sender
                                        options:  (NSDiction-
ary*)slideshowOptions;
        // To start the Slideshow for a PDF:
        – (void)runSlideshowWithPDF:            (NSURL*)pdfURL
                                        options:  (NSDiction-
ary*)slideshowOptions;
        – (void)stopSlideshow: (id)sender;
        – (void)noteNumberOfItemsChanged;
        – (void)reloadData;
        – (int)indexOfCurrentObject;
        – (void)setAutoPlayDelay: (float)seconds;
        @end
        // slideshow option keys:
        //----------------------------------------------------
----------
        extern NSString *const SlideshowDontAutoShowMenuBar;
// boolean
        extern NSString *const SlideshowStartIndex;
// number
        extern NSString *const SlideshowAutoPlayDelay;
// number (in seconds)
        extern NSString *const SlideshowPDFDisplayBox;
// PDFDisplayBox (see PDFKit)
        extern NSString *const SlideshowPDFDisplayMode;
// PDFDisplayMode (see PDFKit)
        extern NSString *const SlideshowPDFDisplaysAsBook;
// boolean
        extern NSString *const SlideshowDebugMode;
// boolean (if YES - only capture main display)
        // notifications a data source may register for:
        //----------------------------------------------------
----------
        extern NSString *const SlideshowDidStopNotification;
        extern NSString *const SlideshowDidChangeNotification;
Export2iPhoto.h
        #import <Cocoa/Cocoa.h>
        @class Export2iPhotoPrivateVars;
        @interface Export2iPhoto : NSObject
        {
                @private
                Export2iPhotoPrivateVars * mPrivateData;
        }
        + (BOOL)canExportToiPhoto;
        + (BOOL)isPhotoFile: (id)file
                allowConversion: (BOOL)allowFormatConversion;
        + (BOOL)exportFile: (id)file;                                   //
file can be: NSFileWrapper
                                                                        //
NSURL
                                                                        //
NSString
                                                                        //
NSImage
                                                                        //
NSArray of any of the above...
        @end
SlideshowCarbon.h
        #ifndef __SLIDESHOWCARBON__
        #define __SLIDESHOWCARBON__
        #if PRAGMA_ONCE
        #pragma once
        #endif
        #include <ApplicationServices/ApplicationServices.h>
        #ifdef __cplusplus
```

-continued

```
        extern "C" {
        #endif
        // callbacks
        typedef CALLBACK_API_C( UInt32 ,        SlideshowNumberO-
fURLsProc ) ( );
        typedef CALLBACK_API_C( CFURLRef ,      SlideshowCopyURLAt-
IndexProc ) (UInt32 index);
        typedef CALLBACK_API_C( Boolean,        SlideshowCanExpor-
tURLAtIndexProc ) (UInt32 index);
        typedef CALLBACK_API_C( void,           SlideshowDidSwitch-
ToIndexNotificationProc ) (UInt32 index);
        typedef CALLBACK_API_C( void,           SlideshowDidEndNo-
tificationProc ) (UInt32 index);
        typedef struct SlideshowCallbacks
        {
                SlideshowNumberOfURLsProc               sNumber
OfURLsProc;
                SlideshowCopyURLAtIndexProc             sCopy-
URLAtIndexProc;
                SlideshowCanExportURLAtIndexProc        sCan-
ExportURLAtIndexProc;
                SlideshowDidSwitchToIndexNotificationProc
sDidSwitchToIndexNotificationProc;
                SlideshowDidEndNotificationProc         sEnd-
NotificationProc;
        } SlideshowCallbacks;
        // Slideshow C APIs
        OSStatus SlideshowRunWithOptions(CFDictionaryRef
options,
                                                SlideshowCallbacks
callbacks);
        OSStatus SlideshowStop( );
        // slideshow option keys:
        //-------------------------------------------------
---------
        extern const CFStringRef kSlideshowDontAutoShow-
MenuBar;    // boolean
        extern const CFStringRef kSlideshowStartIndex;
// number
        extern const CFStringRef kSlideshowAutoPlayDelay;
// number (in seconds)
        extern const CFStringRef kSlideshowDebugMode;
// boolean (if YES - only capture main display)
        #ifdef __cplusplus
        }
        #endif
        #endif /* __SLIDESHOWCARBON__ */
```

Application Programming Interface for objC Client

The following is a description of an application programming inter-face for accessing the functionality of the present invention via an objC client:

SlideshowDataSource Informal Protocol

When a client wants to use the slide show functionality, the client adopts the following informal SlideshowDataSource protocol, using the following two methods:

– (int)numberOfObjectsInSlideshow;	This method is called by the Slideshow framework to find out the total number of images the client wants to display.

-continued

– (id)slideshowObjectAtIndex: (int)index;	Called by the Slideshow framework, this method is used to get an image for a given index. In one embodiment, the 'image' returned can be of one of the following types: NSString (path to image file on disk) NSURL NSFileWrapper NSImage CGImageRef

In one embodiment, the following additional methods are available as well:

– (NSString*)slideshowObjectNameAtIndex: (int)index;	This method is called by the Slideshow framework if the image returned by slideshowObjectAtIndex is of type NSImage or CGImageRef - these two variants do not contain an image name.

-continued

	If the client does return an NSImage or a CGImageRef and does not implement slideshowObjectNameAtIndex, Slideshow displays a generic (Image #) name in the index mode.
– (BOOL)canExportObjectAtIndexToiPhoto: (int)index;	If a client implements this method, Slideshow calls canExportObjectAtIndex ToiPhoto to enable/disable the 'Export to iPhoto' button. If the client does not implement this method -or- iPhoto is not installed -or- an old (incompatible version) of iPhoto is installed, the 'Export to iPhoto' button is not shown in the UI.
– (void)exportObjectsToiPhoto: (NSIndex-Set*)indexSet;	If the user has used the 'Add to iPhoto' action, Slideshow calls the client with an NSIndexSet that contains all indices of the images that should be exported to iPhoto.

The following methods are also provided in the Slideshow API:

+ (Slideshow*)sharedSlideshow;	This class method returns an instance of a Slideshow that can be used to display images or PDFs.
+ (void)addImageToiPhoto: (id)image;	A handy way to add images to iPhoto. It takes as parameter: NSString (path to image file on disk) NSURL NSFileWrapper NSImage CGImageRef NSArray (array of any of the above)
– (void)setDataSource: (id)dataSource; – (void)startSlideshow: (id)sender;	One of two ways to start a Slideshow: setDataSource followed by startSlideshow.
– (void)runSlideshowWithDataSource: (id)sender options: (NSDictionary*) slideshowOptions;	A second way to start the Slideshow. Parameters: dataSource options: SlideshowDontAutoShowMenuBar SlideshowStartIndex SlideshowAutoPlayDelay
– (void)runSlideshowWithPDF: (NSURL*)pdfURL options: (NSDictionary*) slideshowOptions;	Used to start a slideshow with a PDF. Parameters: NSURL of the PDF options SlideshowPDFDisplayBox SlideshowPDFDisplayMode SlideshowPDFDisplaysAsBook
– (void)stopSlideshow: (id)sender;	Used to abort/stop a slideshow.
– (int)indexOfCurrentObject;	Allows the client to find out the index of the currently displayed image.
– (void)setAutoPlayDelay: (float)seconds;	Allows the client to set the auto-play-delay.
SlideshowDidStopNotification	Notification the client may register for. Called when the Slideshow stops (the user did press the cancel button or hit escape)
SlideshowDidChangeNotification	Notification the client may register for. Called when the Slideshow changes from one image to another (either in auto-play or in manual previous/next or via index sheet).

In one embodiment, Export2iPhoto is provided as a helper class used to export images to a photo management application such as iPhoto. It can also be used outside of a running Slideshow. The following methods are provided in the Export2iPhoto API:

+ (BOOL)canExportToiPhoto	Returns YES if iPhoto is installed. It checks also for the correct version of iPhoto.
+ (BOOL)isPhotoFile: (id)file allowConversion: (BOOL) allowFormatConversion;	Returns YES if a given image can be exported to iPhoto. If the image is in a format that iPhoto does not handle directly, the caller has the option to do a format conversion first.
+ (BOOL)exportFile: (id)file;	Exports the given file(s) to iPhoto It takes as parameter: NSString (path to image file on disk) NSURL NSFileWrapper NSImage CGImageRef NSArray (array of any of the above)

Application Programming Interface for Carbon Client

The following is a description of an application programming interface for accessing the functionality of the present invention via a Carbon client:

OSStatus SlideshowRunWithOptions (CFDictionaryRef options, SlideshowCallbacks callbacks);	C API to start the Slideshow with the specified options: kSlideshowDontAutoShowMenuBar kSlideshowStartIndex kSlideshowAutoPlayDelay. callbacks is an array of function pointers that specify the number of images, image location, ... sNumberOfURLsProc: This callback is called by the Slideshow framework to find out the total number of images the client wants to display. sCopyURLAtIndexProc is called to get the image (CFURLRef) at a given index (required). sCanExportURLAtIndexProc This callback is called to find out if the image at a given index can be exported to iPhoto. sDidSwitchToIndexNotificationProc This callback is called to notify client that the user switched to a different image. sEndNotificationProc This callback is called to notify the client that the user has ended the Slideshow.
OSStatus SlideshowStop( );	This callback is called by the client to stop the Slideshow (in case the client quits, the computer is shut down, or the like.)

In one embodiment, the present invention employs techniques described in pending U.S. patent applications cross-referenced above.

In particular, in one embodiment the present invention uses transition effects, user control techniques, and semi-transparency effects, similar to those described in the above-referenced patent applications.

In the above description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the invention. It will be apparent, however, to one skilled in the art that the invention can be practiced without these specific details. For example, the present invention can be applied in any type of application, and for displaying any type of data, including image and non-image files. The particular layouts and user interface elements described herein and illustrated in the Figures are examples only.

Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

Some portions of the detailed description are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

The algorithms and modules presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatuses to perform the method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described

herein. Furthermore, as will be apparent to one of ordinary skill in the relevant art, the modules, features, attributes, methodologies, and other aspects of the invention can be implemented as software, hardware, firmware or any combination of the three. Of course, wherever a component of the present invention is implemented as software, the component can be implemented as a standalone program, as part of a larger program, as a plurality of separate programs, as a statically or dynamically linked library, as a kernel loadable module, as a device driver, and/or in every and any other way known now or in the future to those of skill in the art of computer programming. Additionally, the present invention is in no way limited to implementation in any specific operating system or environment.

It will be understood by those skilled in the relevant art that the above-described implementations are merely exemplary, and many changes can be made without departing from the true spirit and scope of the present invention. Therefore, it is intended to cover all such changes and modifications that come within the true spirit and scope of this invention.

What is claimed is:

1. A computer-implemented method for displaying a set of documents within an email application, the method comprising:

executing a slideshow framework comprising software code for facilitating applications that are independent from the slideshow framework to obtain slideshow functionality provided by the slideshow framework;

executing the email application, wherein the email application is independent of the slideshow framework;

receiving an email message within the email application, the email message comprising a plurality of document attachments;

causing the display of the received email message in the email application;

causing the display of a representation of the document attachments in the email application;

invoking by the email application the slideshow framework, comprising:

responsive to receiving a slide show command, using the slideshow framework to cause the display, within a first user interface that is separate from the email application and without dismissing the email application, of both a slide show comprising the document attachments and a toolbar for interacting with the slide show, the toolbar comprising at least an index sheet element;

responsive to receiving a selection of the index sheet element, using the slideshow framework to cause the display, in a second user interface separate from the email application and without dismissing the email application, of an index sheet comprising a plurality of thumbnail images corresponding to the document attachments; and

responsive to receiving a selection of one of the plurality of thumbnail images during the display of the index sheet in the second user interface, using the slideshow framework to automatically close the second user interface that displays the index sheet and resume the display of the slide show within the first user interface.

2. The method of claim **1**, wherein causing the display of the received email message comprises using the email application to cause the display of a user interface element for activating a slide show.

3. The method of claim **2**, wherein causing the display of the user interface element for activating a slide show comprises causing the display of a slideshow button.

4. The method of claim **2**, wherein using the slideshow framework to cause the display of the slide show comprises using the slideshow framework to cause the display of the slide show responsive to user activation of the user interface element.

5. The method of claim **2**, wherein causing the display of the user interface element for activating the slide show comprises using the email application to cause the display of the user interface element within the context of the email message.

6. The method of claim **2**, wherein using the email application to cause the display of the user interface element for activating the slide show comprises using the email application to cause the display of the user interface element within a header of the email message.

7. The method of claim **1**, wherein causing the display of the slide show comprises using the slideshow framework to cause the display of the slide show within a window associated with the email application.

8. The method of claim **1**, wherein causing the display of the slide show comprises using the slideshow framework to cause the display of the slide show in a full-screen mode.

9. The method of claim **1**, wherein causing the display of the slide show comprises using the slideshow framework to cause the display of the slide show as an overlay superimposed on displayed user interface elements.

10. The method of claim **9**, wherein using the slideshow framework to cause the display of the slide show as the overlay comprises using the slideshow framework to cause the display of the slide show as a semi-transparent overlay superimposed on displayed user interface elements.

11. The computer-implemented method of claim **1**, further comprising:

responsive to receiving the selection of the index sheet element, using the slideshow framework to cause the display of a transition animation prior to causing the display of the index sheet in the second user interface.

12. The computer-implemented method of claim **1**, wherein using the slideshow framework to automatically close the second user interface that displays the index sheet further comprises:

using the slideshow framework to cause the display of a transition animation prior to resuming the display of the slide show within the first user interface.

13. The computer-implemented method of claim **1**, wherein using the slideshow framework to cause the display of both the slide show comprising the document attachments and the toolbar comprises:

using the slideshow framework to cause a sequential display of the document attachments.

14. The computer-implemented method of claim **13**, wherein the sequential display of the document attachments comprises:

causing the display of a first document attachment from the document attachments within the slideshow; and

responsive to a trigger event, causing the display of a second document attachment from the document attachments within the slide show.

15. The computer-implemented method of claim **14**, further comprising:

causing the sequential display of a plurality of document attachments from the document attachments within the slideshow; and

stopping the slide show responsive to receiving a command to stop the slide show.

16. The computer-implemented method of claim 13, wherein the sequential display of the document attachments comprises:

using the slideshow framework to cause the display of a first document attachment from the document attachments within the slideshow; and

using the slideshow framework to cause the display of a second document attachment from the document attachments within the slide show after a predefined amount of time has elapsed from the display of the first document attachment.

17. The computer-implemented method of claim 13, wherein the sequential display of the document attachments comprises:

responsive to receiving user activation of a play command, using the slideshow framework to cause the sequential display of the document attachments;

responsive to receiving user activation of a pause command, using the slideshow framework to remain on a currently displayed document attachment until receiving user activation of another command;

responsive to receiving user activation of a next command, using the slideshow framework to cause the display of a next document attachment from the document attachments; and

responsive to receiving user activation of a previous command, using the slideshow framework to cause the display of a previous document attachment from the document attachments.

18. The computer-implemented method of claim 1, further comprising:

responsive to receiving a command to save a document attachment, adding the document attachment to an image repository associated with a photo management application.

19. The computer-implemented method of claim 1, further comprising:

responsive to receiving user input causing a cursor to hover over a thumbnail image corresponding to a document attachment, causing the display of descriptive information associated with the document attachment.

20. The computer-implemented method of claim 19, wherein the descriptive information comprises at least one of a file name associated with the document attachment, a title associated with the document attachment, a date associated with the document attachment, a file size associated with the document attachment, an author associated with the document attachment, a subject associated with the document attachment, a text description associated with the document attachment, or a compression setting associated with the document attachment.

21. The computer-implemented method of claim 19, wherein causing the display of descriptive information associated with the document attachment:

causing the display of the descriptive information as an overlay that is at least partially superimposed on the thumbnail image.

22. The computer-implemented method of claim 21, wherein the overlay is semi-transparent.

23. A computer program product for displaying a set of documents within an email application, comprising:

a non-transitory computer-readable storage medium; and

executable computer program code, encoded on the medium, for:

executing a slideshow framework comprising software code for facilitating applications that are independent from the slideshow framework to obtain slideshow functionality provided by the slideshow framework;

executing the email application, wherein the email application is independent of the slideshow framework;

receiving an email message within the email application, the email message comprising a plurality of document attachments;

causing the display of the received email message in the email application;

causing the display of a representation of the document attachments in the email application; and

invoking by the email application the slideshow framework, comprising:

responsive to receiving a slide show command, using the slideshow framework to cause the display, within a first user interface that is separate from the email application and without dismissing the email application, of both a slide show comprising the document attachments and a toolbar for interacting with the side show, the toolbar comprising at least an index sheet element;

responsive to receiving a selection of the index sheet element, using the slideshow framework to cause the display, in a second user interface separate from the email application and without dismissing the email application, of an index sheet comprising a plurality of thumbnail images corresponding to the document attachments; and

responsive to receiving a selection of one of the plurality of thumbnail images during the display of the index sheet in the second user interface, using the slideshow framework to automatically close the second user interface that displays the index sheet and resuming the display of the slide show within the first user interface.

24. The computer program product of claim 23, wherein causing the display of the received email message comprises using the email application to cause the display of a user interface element for activating a slide show.

25. The computer program product of claim 24, wherein using the slideshow framework to cause the display of the slide show comprises using the slideshow framework to cause the display of the slide show responsive to user activation of the user interface element.

26. The computer program product of claim 24, wherein causing the display of the user interface element for activating the slide show comprises using the email application to cause the display of the user interface element within the context of the email message.

27. The computer program product of claim 24, wherein using the email application to cause the display of the user interface element for activating the slide show comprises using the email application to cause the display of the user interface element within a header of the email message.

28. The computer program product of claim 24, wherein causing the display of the slide show comprises using the slideshow framework to cause the display of the slide show within a window associated with the email application.

29. The computer program product of claim 23, wherein causing the display of the slide show comprises using the slideshow framework to cause the display of the slide show in a full-screen mode.

30. The computer program product of claim 23, wherein causing the display of the slide show comprises using the slideshow framework to cause the display of the slide show as an overlay superimposed on displayed user interface elements.

31. The computer program product of claim 23, wherein the executable computer program code encoded on the medium is further for:

responsive to receiving the selection of the index sheet element, using the slideshow framework to cause the display of a transition animation prior to causing the display of the index sheet in the second user interface.

32. The computer program product of claim 23, wherein using the slideshow framework to automatically close the second user interface that displays the index sheet further comprises:

using the slideshow framework to cause the display of a transition animation prior to resuming the display of the slide show within the first user interface.

33. The computer program product of claim 23, wherein using the slideshow framework to cause the display of both the slide show comprising the document attachments and the toolbar comprises:

using the slideshow framework to cause a sequential display of the document attachments.

34. The computer program product of claim 33, wherein the sequential display of the document attachments comprises:

causing the display of a first document attachment from the document attachments within the slideshow; and

responsive to a trigger event, causing the display of a second document attachment from the document attachments within the slide show.

35. The computer program product of claim 34, wherein the executable computer program code encoded on the medium is further for:

causing the sequential display of a plurality of document attachments from the document attachments within the slideshow; and

stopping the slide show responsive to receiving a command to stop the slide show.

36. The computer program product of claim 33, wherein the sequential display of the document attachments comprises:

using the slideshow framework to cause the display of a first document attachment from the document attachments within the slideshow; and

using the slideshow framework to cause the display of a second document attachment from the document attachments within the slide show after a predefined amount of time has elapsed from the display of the first document attachment.

37. The computer program product of claim 33, wherein the sequential display of the document attachments comprises:

responsive to receiving user activation of a play command, using the slideshow framework to cause the sequential display of the document attachments;

responsive to receiving user activation of a pause command, using the slideshow framework to remain on a currently displayed document attachment until receiving user activation of another command;

responsive to receiving user activation of a next command, using the slideshow framework to cause the display of a next document attachment from the document attachments; and

responsive to receiving user activation of a previous command, using the slideshow framework to cause the display of a previous document attachment from the document attachments.

38. The computer program product of claim 23, wherein the executable computer program code encoded on the medium is further for:

responsive to receiving a command to save a document attachment, adding the document attachment to an image repository associated with a photo management application.

39. The computer program product of claim 23, wherein the executable computer program code encoded on the medium is further for:

responsive to receiving user input causing a cursor to hover over a thumbnail image corresponding to a document attachment, causing the display of descriptive information associated with the document attachment.

40. The computer program product of claim 39, wherein the descriptive information comprises at least one of a file name associated with the document attachment, a title associated with the document attachment, a date associated with the document attachment, a file size associated with the document attachment, an author associated with the document attachment, a subject associated with the document attachment, a text description associated with the document attachment, or a compression setting associated with the document attachment.

41. The computer program product of claim 39, wherein causing the display of descriptive information associated with the document attachment:

causing the display of the descriptive information as an overlay that is at least partially superimposed on the thumbnail image.

42. The computer program product of claim 41, wherein the overlay is semi-transparent.

43. A system for displaying a set of documents within an email application, comprising:

a slideshow framework comprising software code for facilitating applications that are independent from the slideshow framework to obtain slideshow functionality provided by the slideshow framework;

a computer processor; and

a computer-readable storage medium storing computer program code configured to execute on the computer processor, the code when executed executes the slideshow framework and executes the email application that is independent from the slideshow framework and further causes the slideshow framework to provide UI elements for the email application, the UI elements comprising:

a slide show activation UI element, for using the slideshow framework to cause the display, within a first user interface that is separate from the email application and without dismissing the email application, of both a slide show comprising document attachments included in an email received by the email application and a toolbar for interacting with the slideshow;

an index sheet activation UI element displayed in the toolbar, for using the slideshow framework to cause the display, in a second user interface, separate from the email application and without dismissing the email application, of an index sheet comprising a

plurality of thumbnail images corresponding to the document attachments;

an exit UI element displayed in the toolbar during the slide show, for using the slideshow framework to exit the slideshow and resume the display of the email application; and

wherein the index sheet is configured, responsive to a user selection of one of the plurality of thumbnail images during the display, to cause the slideshow framework to automatically close the second user interface that displays the index sheet and display the slide show within the first user interface.

44. The system of claim **43**, wherein the slide show activation UI element comprises a slideshow button.

45. The system of claim **44**, wherein the slideshow button is displayed within the context of the email message.

46. The system of claim **44**, wherein the slideshow button is displayed within a header of the email message.

* * * * *